REMARKS

The claims remaining in the present application are Claims 1-21. Claims 1, 8, 11, and 14 have been amended. No new matter has been added as a result of these amendments.

EXAMINER INTERVIEW SUMMARY

On April 24, 2003, the Examiner and the Applicants conducted a telephonic interview. Claim 1 was discussed with respect to Mason et al., U.S. Patent No. 5,946,219. A proposed amendment to Claim 1 and a proposed new Claim were discussed. No agreements were reached with respect to the claims. The drawings were discussed. Applicants agreed to send formal drawings to the Patent Office.

DRAWINGS

A submission of formal drawings is being filed herewith. No new matter has been added as a result. Applicants respectfully request that any outstanding objection to the drawings be removed.

CLAIM REJECTIONS

Claims 1-7 and 14-20 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Mason et al., U.S. Patent No. 5,946,219 (hereinafter Mason). The rejection is respectfully traversed.

An embodiment of the Applicants' invention is a computer-implemented method of generating an order of loading data into a programmable device to program the device. The order is automatically generated from a schematic of

the programmable device itself. If a change is made to the schematic of the programmable device itself, a new programming order can be automatically generated, in one embodiment based on the revised schematic of the programmable device itself. Applicants do not understand the cited art to automatically generate an order of loading data into a programmable device from a schematic of the programmable device itself.

CLAIMS 1-7

Currently Amended Claim 1 reads:

> A computer implemented method of generating an order of loading data into a programmable device comprising the steps of:
> a) identifying a plurality of memory cells in a hierarchical schematic representation of a programmable device for which a programming order is desired;
> b) automatically determining a plurality of addresses corresponding to said plurality of memory cells;
> c) automatically determining a plurality of logical names for said plurality of memory cells; and
> d) based on an order in which said plurality of addresses are to be loaded into said programmable device, automatically storing said plurality of logical names for said plurality of memory cells within a data structure within computer readable memory, wherein said data structure describes an order in which to program said programmable device.

Claim 1 recites identifying a plurality of memory cells in a hierarchical schematic representation of a programmable device for which a programming order is desired. Thus, the schematic representation is that of the programmable device, as opposed to a user design implemented therein.

Mason fails to disclose or suggest identifying a plurality of memory cells in a hierarchical schematic representation of a programmable device, as claimed. Rather, Mason discloses a process for implemented a user design in a FPGA.

Mason describes steps that include starting with a design of one or more logic circuits that will be implemented within the FPGA (col. 2, lines 13-15). Mason discloses certain steps taken with respect to this user design. For example, in Figures 4 and 5 Mason discloses entering an initial design (that is to be implemented within the FPGA, as opposed to a design of the FPGA itself). Certain steps may be taken with respect to the user design, which is to be implemented within the FPGA.

Claim 1 further recites that a plurality of addresses and logical names are automatically determined for the plurality of memory cells. Claim 1 further recites, "automatically determining a plurality of logical names for said plurality of memory cells." Claim 1 also recites, "based on an order in which said plurality of addresses are to be loaded into said programmable device, automatically storing said plurality of logical names for said plurality of memory cells within a data structure within computer readable memory, wherein said data structure describes an order in which to program said programmable device." Thus, the order of loading data into the programmable device is determined automatically, based on a hierarchical schematic representation of a programmable device.

Mason may describe generating a configuration bitstream from a design database, using a bitstream compiler (col. 6, lines 23-33). In this fashion, the FPGA may be programmed to implement the user design. However, it is Applicants' understanding that the bitstream compiler is a computer program that was written manually. Such manual generation of a bitstream compiler is very tedious and is subject to errors. Errors in the manual coding of the bitstream compiler can appear

to be errors in the user's design that is to be implemented in the programmable device (Specification, page 1, line 23 – page 2, line 8.). Applicants do not understand the Mason to teach or suggest automatically generating an order in which to program a programmable device using a hierarchical schematic representation of the programmable device, as claimed.

For the foregoing rationale, it is respectfully submitted that Claim 1 is not taught or suggested by Mason. Thus, allowance of Claim 1 is earnestly solicited.

Claims 2-7 depend from Claim 1, which is believed to be allowable. As such, Claims 2-7 are believed to be allowable.

CLAIMS 14-20

For the reasons discussed with respect to Claim 1, Mason does not teach or suggest the limitations of Claim 14. Therefore, allowance of Claims 14 is respectfully requested. Claims 15-20 depend from Claim 14, respectively. As Claim 14 is respectfully believed to be allowable, allowance of Claims 15-20 is respectfully solicited.

CLAIMS 8-13

Claims 8-13 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Mason in view of Varadarajan et al., U.S. Patent No. 5,838,583 (hereinafter Varadarajan). The rejection is respectfully traversed.

Amended Claim 8 reads, in part:

> A computer implemented method of generating an order of loading data into a programmable logic device comprising the steps of:
> a) accessing a data structure comprising a plurality of logical names corresponding to a plurality of addresses in a hierarchical schematic representation of a programmable logic device;
> b) accessing a data structure specifying an order in which said plurality of addresses are to be loaded into said programmable logic device;
> c) automatically ordering said plurality of logical names from step a) based on the order specified in said data structure in step b); and
> d) automatically storing said ordered plurality of logical names from step c) in a data structure within computer readable memory, wherein said ordered plurality of logical names describe an order of loading data into said programmable logic device.

Claim 8 recites "accessing a data structure comprising a plurality of logical names corresponding to a plurality of addresses in a hierarchical schematic representation of a programmable logic device." Claim 8 further recites a limitation of automatically ordering said plurality of logical names from step a) based on the order specified in said data structure in step b). Claim 8 further recites, "automatically storing said ordered plurality of logical names from step c) in a data structure within computer readable memory, wherein said ordered plurality of logical names describe an order of loading data into said programmable logic device." Claim 8 thus recites an embodiment of a computer-implemented method in which an order for loading data into a programmable device is generated from a data structure based on a hierarchical schematic representation of a programmable logic device.

Mason may describe a bitstream compiler. However, for the reasons discussed in the response to Claim 1, Applicants do not understand the bitstream

compiler to perform the automatic ordering or the automatic storing using a data structure that is a hierarchical schematic representation of a programmable logic device, as recited in Claim 8. Thus, Mason fails to teach or suggest the limitations of Claim 8.

Varadarajan fails to remedy this deficiency in Mason. Varadarajan is concerned with the automatic placement and routing of datapath functions (Abstract). For example, the operational flow shown in Figure 2 of Varadarajan begins after the circuit designer has specified the circuit in HDL, performed logic synthesis to produce a netlist, and after basic floorplanning has been done with the chip floorplanner (col. 7, lines 6-10). Varadarajan discloses that the basic schema of the circuit is modified to provide interactive floorplanning (col. 7, lines 20-27). Thus, Applicants understand Varadarajan to disclose modifying a representation of the user's design. However, Varadarajan fails to teach or suggest the automatic generation of an order to load data in a programmable device, based on a schematic representation of the programmable device itself, as claimed.

For the foregoing rationale, Claim 8 is not rendered obvious over Mason in view of Varadarajan. Therefore, allowance of Claim 8 is respectfully requested.

Claims 9-13 depend from Claim 8. As Claim 8 is respectfully believed to be allowable, allowance of Claims 9-13 is respectfully solicited.

CLAIM 21

New Claim 21 recites:

The computer implemented method of Claim 1, further comprising:

e) receiving a modification to said hierarchical schematic representation of said programmable device; and

f) repeating said a) through d) using said modified hierarchical schematic representation of said programmable device to automatically generate a new order in which to program said programmable device.

Claim 21 recites limitations directed to automatically generating a new order in which to program a programmable device, using a modified hierarchical schematic representation of the programmable device. It is respectfully asserted that the cited prior art fails to teach or suggest the limitations of Claim 21. Support for Claim 21 can be found in the Specification on page 3, lines 14-16. Claim 21 is neither taught nor suggested by the cited art.

For example, Mason may describe generating a configuration bitstream from a design database, using a bitstream compiler (col. 6, lines 23-33). In this fashion, the FPGA may be programmed to implement the user design. However, it is Applicants' understanding that the bitstream compiler is a computer program that was written manually. Such manual generation of a bitstream compiler is subject to errors. Moreover, if the architecture of the underlying programmable device changes, then the bitstream compiler must be manually re-written to accommodate the new device, according to Applicants understanding. Applicants do not understand the cited art to teach or suggest automatically generating a new order in which to program a programmable device using a modified hierarchical schematic representation of the programmable device, as claimed.

For the foregoing rationale, it is respectfully submitted that Claim 21 is neither taught nor suggested by the cited art. Therefore, allowance of Claim 21 is earnestly solicited.

CONCLUSION

In light of the above listed amendments and remarks, reconsideration of the rejected Claims is requested. Based on the arguments and amendments presented above, it is respectfully submitted that Claims 1-21 overcome the rejections of record. Therefore, allowance of Claims 1-21 is earnestly solicited.

Should the Examiner have a question regarding the instant amendment and remarks, the Applicants invites the Examiner to contact the Applicants' undersigned representative at the below listed telephone number.

Respectfully submitted,
WAGNER, MURABITO & HAO LLP

Dated: May 6, 2003

Ronald M. Pomerenke
Registration No. 43,009

Address: WAGNER, MURABITO & HAO LLP
Two North Market Street
Third Floor
San Jose, California 95113

Telephone: (408) 938-9060 Voice
(408) 938-9069 Facsimile